REMARKS/ARGUMENTS

Claims 1-20 are pending in the present application. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 103, Obviousness

I.A. Claims 1-5, 8-13 and 16-20 over *Widell* in view of *Lordi*

The Examiner has rejected claims 1-5, 8-13 and 16-20 under 35 U.S.C. § 103(a) as being unpatentable over *Widell* et al., Collision Handling Apparatus and Method, U.S. Patent Application Publication Number 2005/0055490, published March 10, 2005 (hereinafter referred to as "*Widell*") in view of *Lordi* et al., Restoring the State of a Set of Files, U.S. Patent No. 5,857,204, dated January 5, 1999 (hereinafter referred to as "*Lordi*"). This rejection is respectfully traversed.

Widell teaches a data structure being used to store information. A thread is associated with a load vector and a store vector. Each data structure includes a number of bits. There is one bit in the load vector that corresponds to each memory element. There is one bit in the store vector that corresponds to each memory element.

The individual bits are capable of being accessed by a thread. When a thread reads from a memory element, it sets the corresponding bit in the load vector data structure. When a thread writes to a memory element, it sets the corresponding bit in the store vector data structure.

The data structures in the threads are used to handle possible collisions between threads. Collision detection is carried out after a thread has finished executing by means of comparing the data structure of the thread with the data structures of other threads on which the threads depend.

The Examiner states:

> Response to Arguments
>
> 4. Applicant's arguments filed February 27, 2007 have been fully considered but they are not persuasive. Applicant argues that Widell does not teach a first set of data for a first thread associated with a first file, but instead he teaches data structures that are not analogous to files. Applicant further argues that Widell does not describe its data structure as being a file. Lastly applicant argues that the concepts of Lordi cannot be applied to a "data structure." Examiner disagrees.
>
> 5. Widell teaches a first set of data for a first thread associated with a data structure (paragraphs 34, 36 and 39). He then goes on to teach, "many alternative types of data structures for storing information are possible according to the present invention" (paragraph 43). Therefore, it is suggested that a file may be used as the data structure in the Widell application. It is well known in the art that a file may be considered a data structure.

> 6. Nonetheless, the examiner has acknowledged that Widell does not explicitly teach a file system having files, and has relied on the Lordi reference to teach that limitation. Lordi teaches restoring the state of a set of files, in which he teaches a file system having files (column 5 lines 1-7). Therefore, the combination of Widell and Lordi teaches independent claims 1, 8 and 16. Applicant's argument that "Lordi cannot be applied to a "data structure" is incorrect. A file is a "data structure," and Lordi's invention is applied to files.

Final Office Action dated May 8, 2007, page 6.

The Examiner states that *Widell* suggests the use of a file as the data structure of *Widell*. Specifically, the Examiner relies on *Widell*, paragraph 0043, which is reproduced below:

> [0043] The embodiments of the present invention described above comprise data structures in the form of bit vectors for storing information indicative the thread's accesses to the memory. However, many alternative types of data structures for storing this information are possible according to the present invention. The data structures may for instance be implemented as lists to which numbers that correspond to the memory elements are added to indicate accesses the memory elements. Other possible implementations of the data structures include trees, hash tables and other representations of sets.

Widell teaches data structures in the form of bit vectors. Microsoft Computer Dictionary, Fifth Edition, published 2002, defines "vector" as "In data structures, a one-dimensional array—a set of items arranged in a single column or row. *See also* array, matrix." An "array" is defined as "In programming, a list of data values, all of the same type, any element of which can be referenced by an expression consisting of the array name followed by an indexing expression. Arrays are part of the fundamentals of data structures, which, in turn, are a major fundamental of computer programming." A "matrix" is defined as "an arrangement of rows and columns used for organizing related items, such as numbers, dots, spreadsheet cells, or circuit elements..." Therefore, as is made clear by the commonly understood definitions of these terms, a "data structure" is not a "file".

Furthermore, *Widell* provides examples of other data structures that can be used instead of bit vectors. *Widell* states the data structure can be lists to which numbers that correspond to the memory elements are added to indicate accesses of the memory elements, trees, hash tables, and other representations of sets. *Widell* provides examples of other data structures that can be used in place of bit vectors, and expressly does not include a "file" in the list of possible alternatives that can be used in place of bit vector data structures.

The purpose of the data structures is to be used in comparison to handle possible collisions. *Widell* teaches performing the comparison by performing logical AND and logical OR operations on the bits of the data structure in order to handle possible collisions. This comparison is described in more detail in paragraphs 0051-0053, which are reproduced below:

> [0051] The comparison between two data structures to detect collisions is performed differently depending on whether or not the data structures includes separated load and store vectors or a combined load and store vector. If the data structures have separated load and store vectors the comparison between the load and store vectors of an older and a younger thread can be carried out by means of performing the following logical operations bitwise on the bit vectors: old store vector AND (young store vector OR young load vector).
>
> [0052] If the resulting vector contains any bits that are set there is a collision and the younger thread should be rolled back. If the data structures have combined load and store vectors the corresponding logical operation to be performed to check for collisions is an AND-operation between the combined vector of the older thread and the combined vector of the younger thread.
>
> [0053] In an alternative embodiment the comparison to detect collisions is carried out by means of performing the following logical operation bitwise on the bit vectors: old store vector AND young load vector.

Thus, *Widell* teaches carrying out the comparison of the data structures by comparing the bits of the data structure on a bit-by-bit basis. Microsoft Computer Dictionary, Fifth Edition, published 2002, defines "data structure" as "an organizational scheme, such as a record or array, that can be applied to data to facilitate interpreting the data or performing operations on it". Therefore, a data structure is needed so that logical ANDs and logical ORs can be performed on the individual bits of the data structures.

The Examiner stated that *Widell* does not teach a filesystem having files. The Examiner relies on *Lordi* to supply the features that are missing from *Widell*. The Examiner states that *Lordi* teaches a filesystem having files in column 5, lines 1-7, and that it would have been obvious to have modified *Widell* by the teaching of *Lordi*. Applicants respectfully disagree.

Widell and *Lordi* cannot be combined as suggested by the Examiner. *Lordi* teaches files and file operations. The concepts taught by *Lordi* cannot be applied to a "data structure".

Lordi uses the term "file" in a manner that is consistent with the commonly understood definition of the term.

> In the following description, the term "path" will be used in the UNIX sense of a file name. However, adopting this usage is not in any way intended to limit the invention to UNIX embodiments. Further, by common usage, the term "path" may also be used, in a form of shorthand, to refer to the named file itself. Whether a file name or a file is indicated is determined from the context.

Lordi, column 5, lines 36-42.

Lordi clearly teaches a file. Thus, the teachings of *Lordi* apply to "files" and not "data structures", which include bit vectors or arrays.

Because *Widell* does not teach a first thread that is associated with a file, and because *Widell* and *Lordi* cannot be combined as suggested by the Examiner, the combination of *Widell* and *Lordi* does not render Applicants' claims obvious. Therefore, the rejection of claims 1-5, 8-13 and 16-20 under 35 U.S.C. § 103(a) has been overcome.

I.B. Claims 6, 7, 14 and 15 over *Widell* in view of *Lordi* and *Carter*

The Examiner has rejected claims 6, 7, 14 and 15 under 35 U.S.C. § 103(a) as being unpatentable over *Widell* in view of *Lordi* and further in view of *Carter* et al., Remote Access and Geographically Distributed Computers in a Globally Addressable Storage Environment, U.S. Patent No. 5,987,506, dated November 16, 1999 (hereinafter referred to as "*Carter*"). This rejection is respectfully traversed.

Applicants' claims describe wherein the first file comprises an inode page, and wherein the second file comprises a directory page.

The Examiner states:

> **With respect to claims 6 and 14**, Widell as modified teaches claims 1 and 8. Widell as modified does not teach wherein said first file comprises an inode page.
> Carter teaches remote access and geographically distributed computers in a globally addressable storage environment (see abstract), in which he teaches wherein said first file comprises an inode page (column 30 lines 59-61).
> It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have further modified Widell by the teaching of Carter because wherein said first file comprises an inode page would enable Widell's mechanisms to be implemented on a computer network system, having adaptable system configurations for dynamically exploiting distributed network resources and improved fault tolerance (Carter, column 2 lines 55-67).
> **With respect to claims 7 and 15**, Widell as modified teaches wherein said second file comprises a directory page (Carter, column 30 lines 60-61).

Final Office Action dated May 8, 2007, page 5.

The combination of *Widell*, *Lordi*, and *Carter* does not render Applicants' claims obvious because the combination does not teach a first thread that is associated with a file; *Widell* and *Lordi* cannot be combined as suggested by the Examiner; and the combination of *Widell*, *Lordi*, and *Carter* does not teach "change a first set of data for a first thread associated with a first file of said filesystem" in combination with wherein the first file comprises an inode page, and wherein the second file comprises a directory page. Therefore, the combination does not render Applicants' claims 6, 7, 14, and 15 obvious.

II. Conclusion

It is respectfully urged that the subject application is patentable over the cited prior art and is in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: July 9, 2007

Respectfully submitted,

/Lisa L.B. Yociss/ _____

Lisa L.B. Yociss
Reg. No. 36,975
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants